Office of Science
U.S. DEPARTMENT OF ENERGY

Advanced Scientific Computing Research

FY2005 Accomplishments

Collaboratories

The Java CoG Kit helps Researchers World Wide

Gregor von Laszewski *, Argonne National Laboratory

Summary

With the advent of new computing paradigms such as Grid computing it is important to provide researchers with easy to use tools to access the Grid. The Java CoG kit has over the years supported the MICS mission of delivering an easy interface to an advanced compute infrastructure.

In the past, we have seen the development of single computers, mainframes, vector computers, parallel computers, distributed computers, and metacomputers. Most recently, the emergence of the Grid infrastructure helps the scientists to address (1) the ever increasing need to gain access to massive computing power; (2) the recognition that the most powerful computers, while useful for solving complex problems, may be too expensive for an individual computing center or organization to own; (3) the realization that connecting such resources as part of a persistent infrastructure is a challenging management issue; (4) the need for integrating noncomputational resources into the Grid; and (5) the importance of standards to promote interoperability among participants.

The Java CoG Kit has been one of the first toolkits that made Grid computing accessible for scientists. Simon Cox from the GEODISE project statet: "**the Java CoG Kit is an essential tool to transition the Grid from a *useful* resource to a *usable* resource"**. GEODISE is a research project hosted in the UK involving multi-disciplinary teams working to build a state-of-the-art engineering design, search, and optimization tool for fluid dynamics.

At Supercomputing 2004 the Java CoG kit was used to interface components of the Linked Environments for Atmospheric Discovery.(LEAD) to the Grid with the goal to develop better than real-time forecasts of mesoscale weather events, such as tornadoes. Other application support is often through indirect support of third party projects such as Pegasus as Ewa Deelman reports. The Java CoG Kit has allowed Pegasus to access the Grid easily in order to support applications including the National Virtual Observatory, the SLOAN Digital Sky Survey, Bioinformatics and High Energy Applications. Most recently, the Java CoG kit has enabled a large number of Portals based on the OGCE software to access Grid technologies. Technologies developed by the Java CoG Kit have also enabled to build compute services that are reused as part of the DOE Collaboratory for Multi-Scale Chemical Sciences (CMCS) and the application called Active Thermochemical Tables have ushered in a new era of validated chemical reference data where all the pertinent experiments and computations can be considered by all the experts. A 50 year old question of the vaporization enthalpy of graphite, a thermochemical reference value for countless computational and experimental studies, has been resolved by an IUPAC

* 630 252 0472, gregor@mcs.anl.gov

Task Group empowered by CMCS capabilities. The new value for the enthalpy of formation of carbon atom in the gas phase is now more than twice as accurate. These new capabilities also enabled a group at DOE's Argonne National Laboratory to fully confirm a recent revision of the enthalpy of formation of the pivotal combustion and atmospheric radical, hydroxyl (OH), and to further reduce its uncertainty by a factor of ~6.5, thus removing a potential source of uncertainty in current chemical models.

One of the other highlights in the FY2005 included the reward of the **best poster award at Supercomputing 2004** among 109 other entries hence supporting MICS quest for the development of state of the art research contributions in the field of supercomputing. The Java CoG Kit project was also successful in another mission of MICS targeted in to the education of future researchers, One of its summer undergraduate students has received the best undergraduate project award from the Polytecnic NY.

To depict the use of the Java CoG Kit we have mapped in Figures 1 and 2 the 153 traceable downloads of the Java CoG Kit for the United States and the 749 traceble downloads outside the US from November 2004. However, we have to mention that the number is larger as we did not include the about 2000 non traceable downloads and downloads that result through third party distributions such as the reuse of the CoG Kit within the many projects using the Globus Toolkit, Pegasus, OGCE, or others.

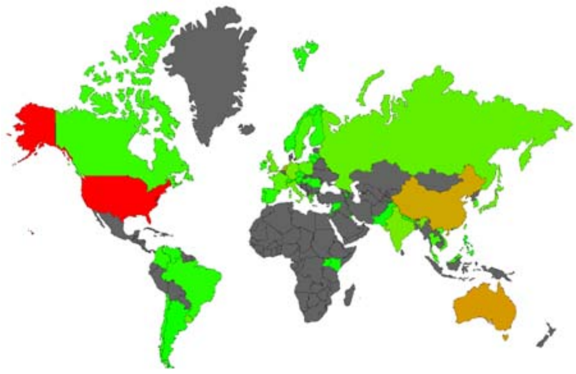


Figure 1. Recent traceable downloads by countries. The colors indicate quantity. Red > 150 > Brown > 90 > Dark Green >20 > Green > 0.

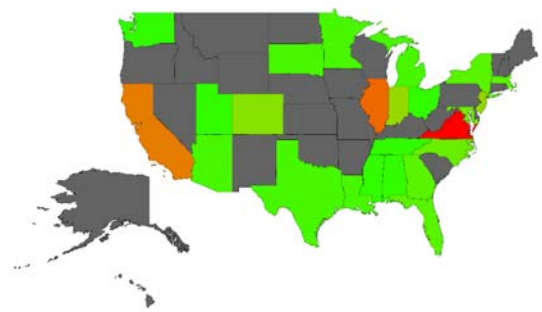


Figure 1. Recent traceable downloads of the Java CoG Kit by states. The colors indicate the quantity: Red > 28 > Brown > 24 > Dark Green > 8 > Green > 2 > Light Green > 0.

Most recently the Java CoG kit has been enhanced to provide scientific workflows and bridge to a number of significantly different Grid middleware implementations including the newest versions of the Globus Toolkit.

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

CoG Kits Support Portals for the Scientific Community

From Einstein to Zweistein with Portals

Gregor von Laszewski,* Argonne National Laboratory

Summary

Scientific portals help establish easy access to complex information, data, and infrastructure as used of the scientific discovery process. The Java CoG Kit has demonstrated that it provides valuable features to build Grid-based scientific portals.

Collaboration is an essential part of the scientific discovery process. Grids can be viewed as a tool to help in improving collaboration through shared virtual organizations. In order to make this happen, agreements must be put in place policies that govern the extent to which sharing of resources is allowed. Such sharing is not limited to the instantiation of a program: considerable human interaction is necessary to agree on the policies under which such automated systems operate. Once these policies are in place and systems on the Grid are developed and deployed, scientists hope to enable what we term the *Zweistein effect*, a play on the name Einstein. Instead of having a single scientist working alone in his laboratory, we strive to develop an infrastructure that allows the unique capabilities of a number of scientists to be integrated. What if we could create a Zweistein effect, a Dreistein, or even an N-stein, by having N researchers communicate with each other? (Eins, zwei, drei are the German numbers one, two, three.)

The need for such an infrastructure is obvious. Computational scientists often develop large models and codes intended to be used by larger user communities or for repetitive tasks such as parametric studies. Lowering the barrier of entry for access and sharing to these codes is often a technical and sociological challenge. Portals help bridge the gap because they are well-known interfaces enabling access to a large variety of resources, services, applications, and tools for private, public, and commercial entities, while hiding the complexities of the underlying software systems to the user. But developing portals is a challenging undertaking: it interfaces not only with the scientific discipline but also to the often complex and quickly changing computational resources and software tools.

For several years the Java Commodity Grid (CoG) Kit has helped in providing an easy and robust bridge to the fast-changing infrastructures developed for the scientists. The CoG Kit has especially helped in making job submission, data transfers, and secure authentication easier for the portal developer community. The libraries developed by the Java CoG Kit are reused in many scientific Grid portals, not only within DOE projects, but also in agency overarching within NSF, and NIH. As such the CoG Kit is a valuable asset to portal developers to expose scientific results in the community and building new collaborative

* Mathematics and Computer Science Division, (630) 252 0472, gregor@mcs.anl.gov

opportunities between scientists. Some of the portals that benefit from our software include the Earth System Grid (Fig. 1), the Fusion Grid Portal (Fig. 2), and the Linked Environments for Atmospheric Discovery (Fig. 3).

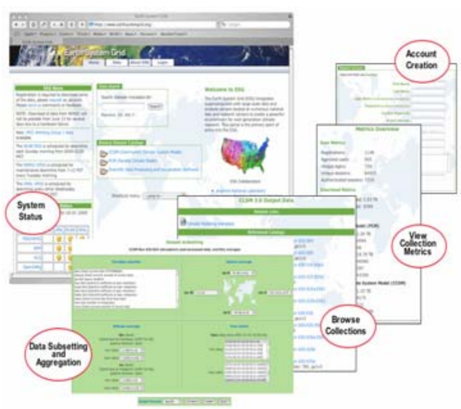


Figure 1: Earth System Grid (ESG): enables the community to access to climate simulation results distributed across DOE and NSF supercomputers. The CoG Kit is used to provide the basic security mechanisms Over 1,200 users are registered and have access to climatology simulations and over 75 TB of data in over 400,000 files.

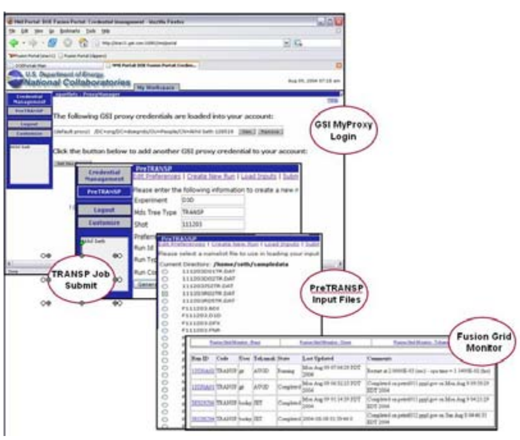


Figure 2: The Fusion Grid Portal provides a single view of the distributed data collections, and simulations. It is based on a portal toolkit that relies on the Java CoG Kit basic functionality.

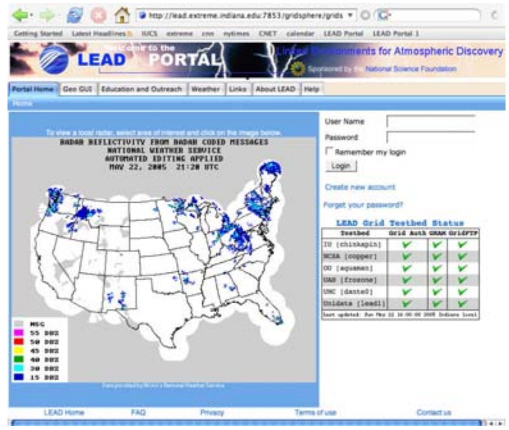


Figure 3: The Linked Environments for Atmospheric Discovery (LEAD) Portal allows users to manage data-driven, high performance weather simulations.

References

[2] Java CoG Kits, http://www.cogkit.org
[1] Grid Portal Architectures for Scientific Applications, M. P. Thomas, J. Burress, G. Fox, D. Gannon, G. von Laszewski, K. Jackson, R. Moore, M. Pierce, B. Plale, R. Regno, D. Schissel, A. Seth, SciDAC 2005 Meeting.

For further information on this subject contact:

Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

Python Commodity Grid Kit (pyGlobus)

Keith R. Jackson, Lawrence Berkeley National Laboratory

Summary

The Python CoG Kit, or pyGlobus, provides easy access to the full functionality of the Globus Toolkit™ from the Python programming language. Python excels at being a lightweight scripting language that is also suitable for larger, more sophisticated tasks The CoG Kit aims to make Python the scripting language for the Grid, facilitating rapid grid application development. The main emphasis of the Python CoG Kit project in the past year has been on supporting our scientific users, and developing a collaborative visual programming environment to support complex, collaborative, scientific workflows.

The Python CoG project aims at increasing scientific usage of the emerging national middleware infrastructure that DOE researchers have pioneered by providing access to the Globus Toolkit from the high-level, object-oriented language Python. Python is a high-level interpreted language that supports a rapid application development cycle. Python's minimal syntax makes it an ideal language for usage by non-computer scientists. By enabling scientists to focus less on computer science details, pyGlobus allows scientists to focus on their science.

One example of a successful usage of pyGlobus is the NSF funded Laser Inferomneter Gravitational-Wave Observatory (LIGO). LIGO researchers have used the Python CoG Kit to replicate over 5-TB's of data to more then 40 sites on three continents.

LIGO, the Laser Interferometer Gravitational-Wave Observatory, is a facility dedicated to detecting cosmic gravitational waves -- ripples in the fabric of space and time -- and interpreting these waves to provide a more complete picture of the universe. LIGO consists of two widely separated installations -- one in Hanford, Wash., and the other in Livingston, La. -- operated in unison as a single observatory. Data from LIGO will be used to test the predictions of General Relativity -- for example, whether gravitational waves propagate at the same speed as light, and whether the graviton particle has zero rest mass.

Because gravitational waves have never been directly detected (although their influence on distant objects has been measured), LIGO is conducting blind searches of large sections of the sky and producing an enormous quantity of data -- almost 1TB a day -- which requires large-scale computational resources for analysis.

The LIGO Scientific Collaboration (LSC) scientists at 41 institutions worldwide need fast, reliable and secure access to the data. To optimize access, the data sets are replicated to computer and data storage hardware at nine sites: the two observatory sites plus Caltech, MIT, Penn State, the

(510)-486-4401, KRJackson@lbl.gov

University of Wisconsin-Milwaukee (UWM), the Max Planck Institute for Gravitation Physics/Albert Einstein Institute in Potsdam, Germany, and Cardiff University and the University of Birmingham in the United Kingdom.

The data distribution tool used by the LSC DataGrid is the Lightweight Data Replicator (LDR), which was developed at UWM as part of the Grid Physics Network (GriPhyN) project. LDR is built on a foundation that includes the Globus Toolkit, Python, and pyGlobus, an interface that enables Python access to the entire Globus Toolkit. LSC DataGrid engineer Scott Koranda describes Python as the "glue to hold it all together and make it robust."

PyGlobus allowed LIGO scientists to develop their experimental infrastructure faster then otherwise possible, and allowed the scientist to focus on their data analysis.

In addition to support scientific projects such as LIGO, we have been developing a Visual Composition Environment, ViCE, to support the collaborative development and execution of complex scientific workflows.

Scientific projects today are frequently large collaborations amongst geographically and organizationally distributed teams. ViCE is designed to support scientists collaborating over a visual description of their workflow. A workflow is represented as a set of nodes and links. The nodes represent actions, such as querying a protein sequence database, and links represent the data transfer between nodes. By dragging and dropping a series of domain specific nodes onto a palette, a scientist can construct a complete workflow. Figure 1 shows a typical visual workflow description from biology. The biologists are searching several protein sequence databases looking for a likely match to a newly sequenced protein.

ViCE supports collaboration by allowing multiple groups to have the same view of the changing workflow description. They can use integrated chat tools to discuss the workflow. Future versions will support the collaborative editing of the visual workflow description.

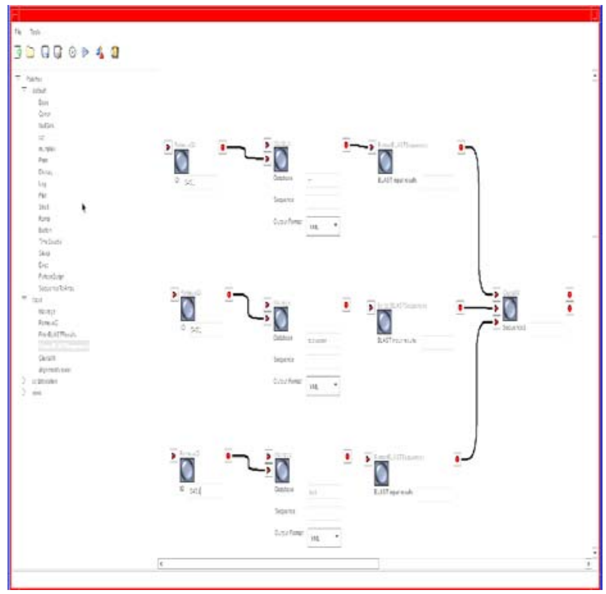


Figure 1. A typical ViCE workflow that shows a scientist searching 3 protein sequence databases (nr- the main protein sequence database, topsecret- an example "top secret" proprietary data, and pdb- the protein structure databank") using BLAST, then extracting the result sequences and aligning them using the ClustalW applicaiton.

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

eServices Project

Keith R. Jackson*, Lawrence Berkeley National Laboratory

Summary

The eServices project is developing the next generation of Grid middleware to support scientific applications. The most recent Grid standards, i.e., the Web Service Resource Framework, WSRF, and Web Service Notification, WS-N, are based on industry standard Web Services. A Web Service is simply any service that describes its interface in a standard format, based on XML, and is accessible via standard Web protocols. The usage of standard protocols enables the scientific world to leverage the significant corporate investment in Web Service Infrastructure.

The eServices project aims at increasing scientific usage of the emerging national middleware infrastructure that DOE researchers have pioneered by providing an implementation of the WSRF family of specification in the Python programming language called pyGridWare. Python is a high-level interpreted language that supports a rapid application development cycle. Python's minimal syntax makes it an ideal language for usage by non-computer scientists. It also easily supports binding together C/C++, and FORTRAN codes and exposing them through a thin Python interface. By enabling scientists to focus less on computer science details, pyGridWare allows scientists to focus on their science.

We have released the first versions of pyGridWare as part of the Globus Toolkit™ 4.0 release. This is an important milestone towards supporting the transition from a pre-Web Service Grid to a WSRF based Grid for the Python Grid community.

PyGridWare supports the development of WSRF based applications and services, and interoperates with the Java and C implementations from Argonne National Lab, the .Net version from University of Virgina, and the Perl implementation from University of Manchester.

In addition to support for developing WSRF applications from scratch, we have developed a tool to automatically wrap an existing command-line application to expose it as a WSRF service. This allows a scientist to take an existing application that they run locally, and expose it as a Grid Service that is accessible over the network. By allowing existing applications to be easily converted into Grid Services we hope to leverage the significant investment DOE has made in high-performance codes, such as those developed under the SciDAC ISIC's while still exposing these applications as Grid Services as part of the emerging national middleware infrastructure.

As part of our ongoing work, we will continue to develop pyGridWare as the underlying standards change. This is important to ensure interoperability between all of the different WSRF implementations. In addition, to support the needs of the high-

*(510)-486-4401, KRJackson@lbl.gov

performance scientific community, we will be working to increase the performance of the underlying toolkit.

Another important area of ongoing work is supporting our users transitioning from a pre-Web Service Grid based on the pyGlobus project also developed here, to a Web Service based Grid based on pyGridWare. The DOE funded Access Grid is one group that is in the process of transitioning to pyGridWare for their infrastructure. The Access Grid is s an ensemble of resources that enables group-to-group collaborations amongst interdisciplinary scientists. It supports high-end video and audio conferencing amongst other collaborative tools. PyGridWare provides an efficient means for authentication and secure communication between different rooms.

Another important application is the Laser Interometer Gravitation Observatory, LIGO. The current LIGO infrastructure for moving data amongst the 40+ sites involved in the collaboration is based on our earlier work on pyGlobus. We have been working closely with LIGO engineers to help them begin the transition to the next generation of Grid middleware.

For further information on this subject contact:

Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

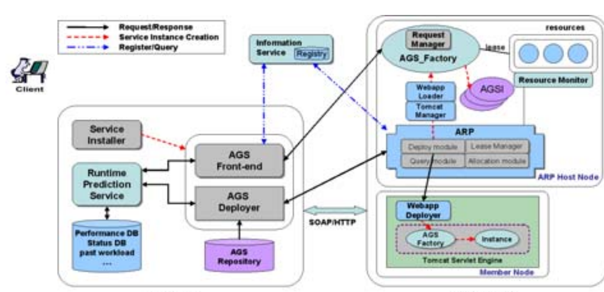
Securing Cyberspace with Community Services

Jon B. Weissman*, Department of Computer Science
University of Minnesota

Summary

High-performance distributed applications are increasingly multidisciplinary, collaborative, and dynamic. Grid computing which harnesses resources at multiple sites through common access protocols has a great potential to be a future hosting platform for such applications. Recent efforts have focused on standardizing Grid interfaces to enable better interoperability in the form of Grid services. However, Grid services must cope with the uncertainty of service demand and the availability of network resources. **This is particularly true in collaboratories in which services and the resources upon which they are hosted are highly shared.** If the Grid service provider wishes to provide efficient and robust services, then he/she must develop complex adaptation code as there is little existing infrastructure to utilize. This increases the time-to-deployment which in turns increases the time-to-solution for applications that require such services.

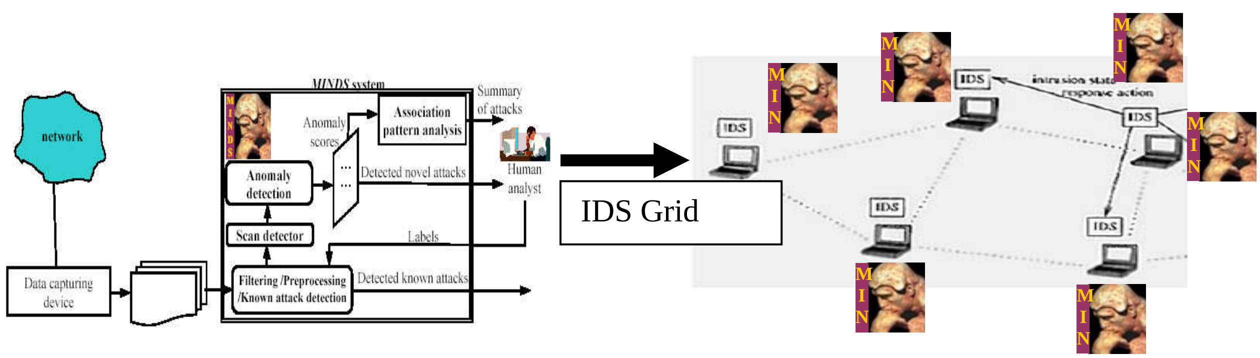
The Community Service project is developing re-usable middleware and system components that will fill the gap between Grid services and the dynamics of the collaboratory environment. The project is focusing on high-end services which encapsulate parallel and distributed computing, and may access large datasets. Such services may represent kernel codes that have become de-facto standards for production applications. When such codes are transformed into services, several issues arise due to network dynamics: (1) how should resources be allocated to multiple concurrent requests for a high-end service? (2) where should a high-end service be hosted, re-hosted, or replicated? (3) to which service replica should a request be sent? and (4) how to make services appear robust? Middleware is being developed to address these problems. For item (4), we have developed a dynamic deployment mechanism (shown below). Unreliability stems from two sources: service demand overload and host node and network failure. Our middleware automatically re-deploys overloaded services to new resources pools to accommodate high demand, and automatically detects and responds to host node and network failure by re-deploying a failed service on an available resource pool. Collectively, these allow the end-user application to receive efficient and reliable results from a service irrespective of load or failure.



* 612-626-0044, jon@cs.umn.edu

One of the exciting applications of this technology is cybersecurity.
Network attacks such as intrusions are increasing with alarming frequency and the discovery of attacks is often done "after-the-fact" by analysis of network logs. Single-site intrusion detection systems such as MINDS (at the University of Minnesota, Kumar, PI) work well but cannot catch sophisticated attacks that involve several sites (using hacked machines to launch subsequent attacks). Researchers at the University of Illinois (Robert Grossman), University of Florida (Sanjay Ranka), and the University of Minnesota (Vipin Kumar), are exploring the concept of distributed network intrusion detection in which network data from multiple sites can be combined and scanned to detect a higher number of attacks, and may be possibly be used to predict future attacks (see below). This kind of collaborative intrusion detection is utilizing the dynamic deployment capability of Community Services. In particular, we package existing single-site detection systems (e.g. MINDS) into dynamic Grid services that can be deployed automatically.

If a site wishes to join an "IDS Grid", it can use Community Service tools to locate the IDS service in a registry and automatically deploy it on a machine in their local network. It can then learn about the other sites (and vice-versa) and subscribe to their intrusion alerts and publish alerts to the IDS Grid. One of the powerful applications is that collective alert information may be used to identify threats that were ignored previously. For example, an event that is below threshold for a single site, but appears at multiple sites, may be signal that it poses a real threat. The Community Service middleware and tools makes it easy for sites to join an IDS Grid. The advantage is that the larger the IDS Grid, the greater the source of shared data that can be used to detect and predict cyber-attacks, as an attack on one site, may presage similar attacks on other sites. Future work is to continue to apply Community Service technology to other DOE collaboratory applications such as in the FusionGrid.



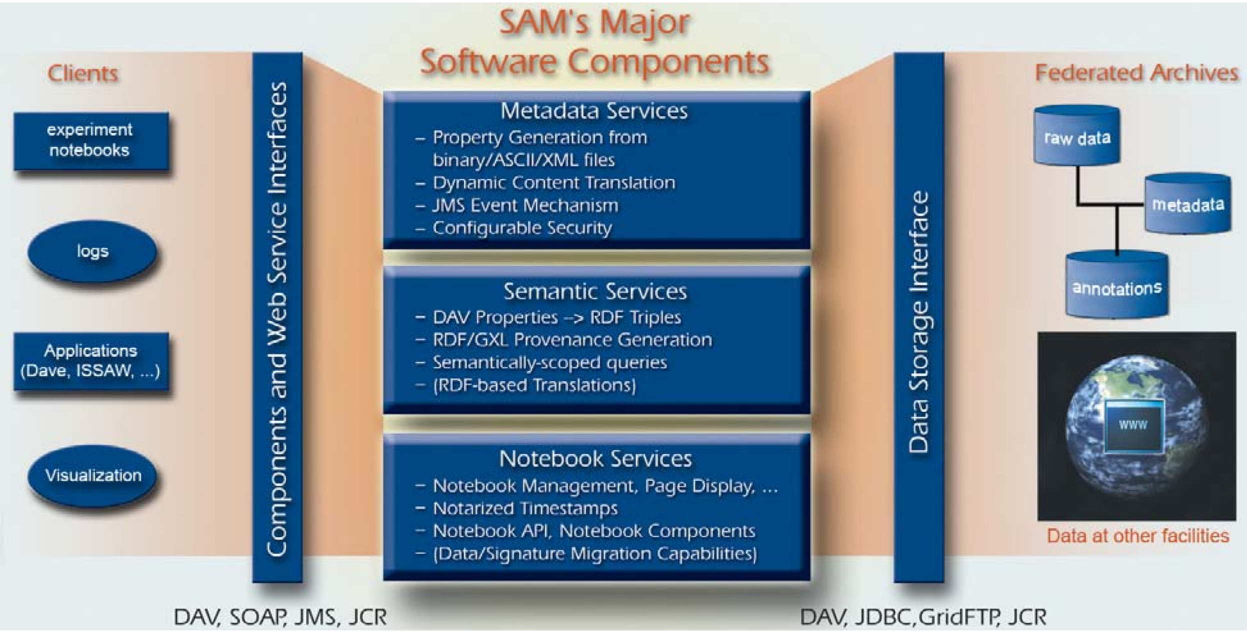
For further information on this subject contact:

Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

Advanced Scientific Computing Research
FY 2005 Accomplishment

2005 Highlight in Scientific Annotation Middleware

Jim Myers, PNNL and Jens Schwidder*, ORNL



The Scientific Annotation Middleware (SAM) system (http://www.scidac.org/SAM) provides significant advances in research documentation and data provenance tracking required for effective management and coordination of the complex, collaborative, cross-disciplinary, compute-intensive research enabled through the Scientific Discovery through Advanced Computing (SciDAC) initiative. The SAM project is helping the Spallation Neutron Source (SNS) to provide users with tools to manage and annotate data produced in experiments, analysis, and simulations and create extensive electronic research records.

The Scientific Annotation Middleware (SAM) is being developed by researchers at Oak Ridge National Laboratory and Pacific Northwest National Laboratory as a layered set of components and services supporting the creation and use of annotation metadata about data objects and the semantic relationships among them. SAM is built upon web, Grid, and semantic web technologies as well as the successful DOE2000 electronic notebook research.

The SAM project is addressing the needs of next-generation Grid-based scientific research to federate data and metadata, track provenance, document research processes, and expose such information to a wide range of applications, agents, workflow services, notebooks, and scientific portals. The key concept behind SAM is a “schema-less” data store that can accept arbitrary input and the use of dynamically registered translators to map data and metadata into the formats, schemas, and ontologies expected by applications and underlying data repositories. This allows researchers to capture records-related information using an arbitrary combination of tools and to later define how this

* (865) 576-7928, schwidderj@ornl.gov

information should be translated into forms interpretable in other contexts, e.g. into the input format required by a collaborator's software, the schema of a community database, or that of a records-management tool or automated virtual-data/workflow system. In the SAM model, it becomes possible to view all of the recorded information via a single interface/protocol while simultaneously defining limited views of that data that conform to the conceptual models of particular applications, groups, institutions, or communities.

SAM is being used to develop a new generation of electronic notebook software that can take full advantage of the richer, federated records and the semantic capabilities. It will serve as the basis for a variety of mechanisms like third party annotation and peer-evaluation in addition to advanced group notebooks and notebook components embedded in other tools.

The SNS software project is looking at SAM technologies and experiences for annotating instrument logs. The logs will be customized based on the requirements of the different instruments. SAM will provide SNS users with the ability to create rich experiment records automatically capturing relevant information like instrument configuration and experiment parameters and allowing custom annotations. The SNS will also require provenance tracking and ways of annotating data with user defined metadata to support knowledge discovery and comprehension. The SNS software development is coordinated internationally with other neutron science facilities. The translation and semantic capabilities provided in SAM can provide mechanisms for bridging the gaps between the different facilities.

The TeraGrid is a partner in the SNS software development and will provide the SNS with computational resources as well as the infrastructure and services for other research partners to access data and services at the SNS. Since SAM is true middleware – it can be integrated with Grid technologies. It can be configured to use existing security and data storage services. Authentication can be performed using any Java Authentication and Authorization Service (JAAS) provider, including external username/password databases, public key certificates/Grid security infrastructure. Data and metadata can be stored in files, remote databases, or repositories accessed via GridFTP.

The SAM team also participates in several Grid Forum standardization efforts including the Data Format Description Language (DFDL), Grid Computing Environments (GCE), Grid Information Retrieval (GridIR), and Relational Grid Information Services (RGIS) working group.

The SAM source code has been moved to SourceForge (sourceforge.net) to invite community contributions to the development effort and create a basis for the long-term support of the SAM software.

SAM already provides powerful capabilities for data integration, metadata/pedigree management, and free-form annotation. As the project continues, we anticipate a variety of improvements that will enable additional user groups. Version 2.1 of SAM has been release under an open source license in March 2005. Version 2.2 will be released in November 2005.

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

Storage Resource Management for Distributed Data Applications

PI: Arie Shoshani* (LBNL)
CoPIs: Don Petravick (Fermilab), Alex Sim (LBNL)

Summary

Storage Resource Managers (SRMs) are middleware components whose function is to provide dynamic space allocation and file management on shared distributed storage systems. This effort supports the MICS mission in providing the technology needed to manage the rapidly growing distributed data volumes, as a result of faster and larger computational facilities. Over the last year, we deployed our SRMs in real production projects including multiple High Energy Physics experiments as part of the Particle Physics Data Grid (PPDG) project, and the Earth Science Grid (ESG) project, and the Scientific Data Management ISIC. SRMs are used in these projects for space management, robust file movement, and streaming data for analysis. In addition, we continued the coordination of an international effort to standardize SRM interfaces, and developed a powerful second generation standard specification.

One of the goals of the Storage Resource Management project was to develop a standard specification against which multiple implementations can be developed. This approach proved to be a remarkable and unique achievement, in that now there are multiple SRMs developed in various institutions around the world that interoperate.

The SRM functional specification effort was initiated by the Scientific Data Management Group (SDM) at LBNL. Because the SDM group is also participating to deploy SRMs in two of the National Collabratory projects, Earth System Grid and the Particle Physics Data Grid, SRM has developed into an internationally coordinated effort between several DoE laboratories including LBNL, Fermilab and TJNAF, as well as European institutions including CERN and RAL in the UK. This coordinated effort has resulted in the adaptation of the standard specification, and the development of multiple SRM middleware components in various institutions around the world to interface to their specific storage systems. This approach is particularly essential for providing distributed access to complex Mass Storage Systems (MSSs). SRMs were even developed for legacy MSS systems (e.g. at NCAR) which enables them to be accessed from the Grid. The concept of interoperability is illustrated in Figure 1, where the same client can interact with multiple SRMs implemented over existing diverse storage systems. The SRMs can also communicate with each other in a peer-to-peer fashion request and move files.

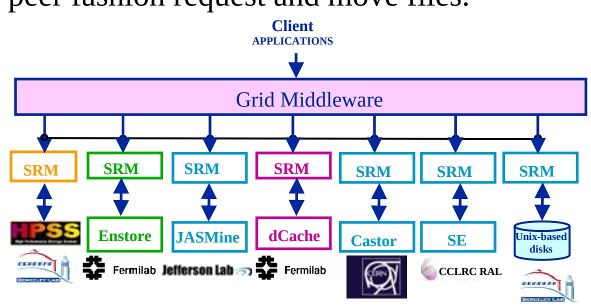


Figure 1: SRMs facilitate access uniformity of diverse distributed storage systems

* 510.486-5171, shoshani@lbl.gov

SRMs have been used in production by several facilities including BNL, NERSC, Fermilab, CERN, and JTNAF. Some are being used for access to files or for storing files in remote systems, and some are used for intensive data movement between storage systems. For example, The BNL to NERSC setup using SRMs that access HPSS, takes advantage of robustness features of the SRMs. They use the Berkeley-SRMs to move about 10,000 files per month (about 1 GB each) in an automated fashion. This arrangement, called a DataMover, is also being used by the Earth Systems Grid to move robustly a large volume of simulation production data from NERSC to NCAR, as well as ORNL to NCAR. The benefits are great reduction in the error rates, and essentially eliminating the human effort previously required. For example, a 50 fold reduction in error rates (from 1% to 0.02%) was achieved in routine file replication of the STAR experiment.

Another example of a successful deployment in the SRM-dCache developed at FNAL. It is widely deployed for use in the CMS project, and it interoperates with the SRM-Castor at CERN. This effort demonstrates the usefulness of SRMs by achieving sustained SRM-to-SRM managed transfers from Castor to FNAL dCache and onto tape at a rate between 40 and 60 MB/s. In addition, high rate, CERN disk to FNAL disk utilizing SRM clients were performed from approximately 150 nodes requested data from CERN. This maxed out at ~700 MB/s.

SRMs have also been used for projects in the Scientific Data Management ISIC. In one application, called a GridCollector, SRMs were used in combination of an efficient indexing method to greatly speed up the analysis of high energy data. In another project, SRMs are used to facilitate data intensive applications that use MPI-IO to transparently access files on remote storage systems, including mass storage systems. This is illustrated in Figure 2.

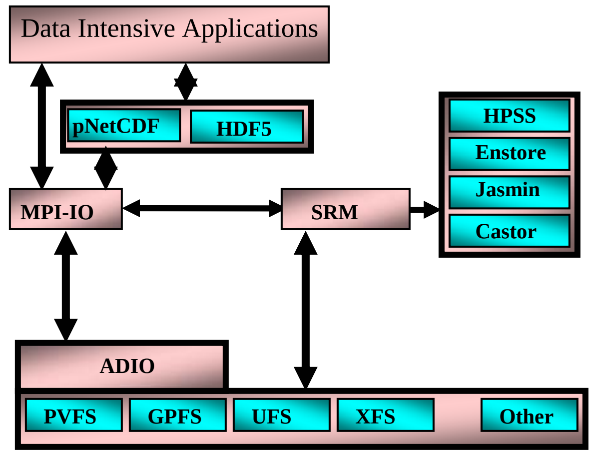


Figure 2: SRMs use for transparent access in data intensive applications.

The dream of having multiple, diverse, distributed storage system that interoperate was fulfilled by this effort, thanks in great part to the goodwill and commitment of the international participants. The SRM standard is now widely adopted by various international efforts. Although the SRM specification has become a de-facto standard, there is an ongoing effort to standardize this functional specification through the Global Grid Forum (GGF). This activity is facilitated by an international SRM-collaboration that continues to enhance the SRM standard specification based on experience and new requirements.

For further information on this subject contact:

Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

Advanced Scientific Computing Research
FY 2005 Accomplishment

Providing a Pervasive Collaborative Computing Environment (PCCE)

D. Agarwal*, D. Gunter, D. Konerding, M. Rodriguez, Lawrence Berkeley Laboratory

Summary

This paragraph should contain a short(!) executive summary which identifies the objectives of the project, its relevance to the mission of MICS and the advances made during FY2005.

Collaboration among scientists spread worldwide is critical to modern big science. Projects like the Fusion Collaboratory, the Compact Muon Solenoid (CMS) detector, the Atlas detector, and the Earth Systems Grid (ESG) need to coordinate the activities of a worldwide collaboration.
Collaborators need to be abletools which allow them to find and contact other participants, chat, leave notes, share and edit documents, track current workflow, and activate videoconference sessions, dynamically as needed. In addition, to support these global collaborations, the tools for this shared space must, where necessary, accommodate asynchronous interactions between participants not present within the space at the same time.

The purpose of this PCCE project is to develop and deploy software tools that support a persistent collaborative "space" within which participants can rendezvous and interact with each other. Persistence and security play a major role in acceptance of collaborative technologies and are addressed in the fundamental system. In this project we are working across the range of collaboration tools including messaging, Access Grid, wikis, blogs, and document sharing. Also, many scientific collaborations are organized around common computational goals which are most easily expressed within a workflow-management environment.

Progress to Date

Our concentration during the PCCE project has been on building a collaboration framework and establishing the base set of tools required to establish an easy to use and persistent environment. We have pursued this through use of existing tools when possible, adapting existing tools when necessary, and building new tools as needed to complete the environment.
The foundation of the PCCE environment consists of presence, contact, text messaging, and workflow tools. These capabilities allow users to collaborate securely by running a lightweight interface that requires no additional hardware and can be accessed via the web. It can be run continuously on each user's desktop.
The PCCE environment leverages existing and emerging infrastructure, such as grid services, the Jabber XMPP protocols, Web services, wikis, blogs, and Access Grid capabilities.

Accomplishments FY'05

In the last year, we have completed our python Jabber library called Bajjer and it is being incorporated into the Access Grid as a

* (510)-486-7078, E-mail: DAAgarwal@lbl.gov

replacement for the existing chat. This will significantly improve the flexibility and usability of the chat feature. We have also continued to improve and further develop our Jabber archiver to improve robustness and search features. The Jabber archiver is now fully functional and in regular use. This year we have also begun to leverage the existing TWiki Wiki and WordPress Blog open source software projects to provide wiki and blog functionality. Since these projects are open source, we have been able to modify these tools to fit within our framework. Figure 1 depicts some of these tools.

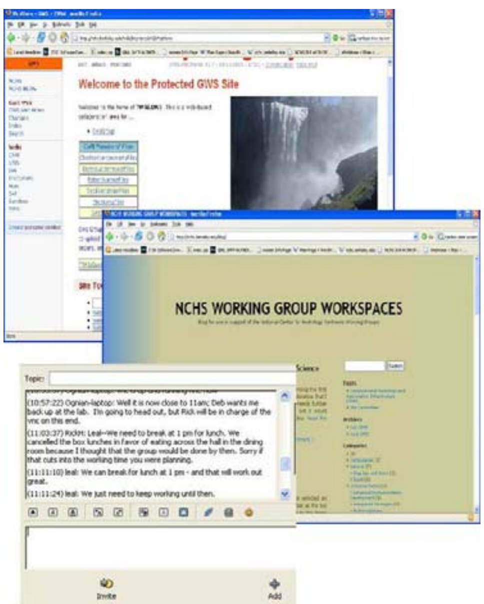


Figure 1: Collaborative tools within PCCE

In order to support scientific collaborations centered on large-scale computing activities, shared collaborative control and monitoring of these tasks is needed. We have concentrated this year on implementing workflow-management tools which allow shared management of a collaboration's computing activities. The workflow environment we are building is called ViCE. ViCE has been developed in python and provides an intuitive visual interface. It is built with distributed computing, grid, and web services as core capabilities in the system. ViCE is currently being used to model and run a structural biology problem currently. Figure 2 shows a structural biology workflow representation in ViCE.

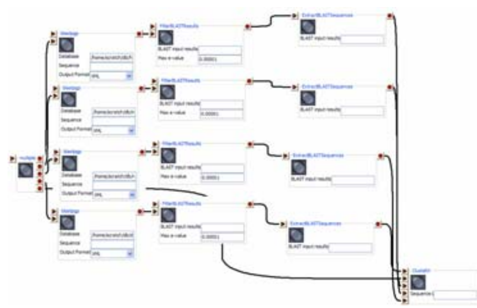


Figure 2: ViCE workflow

Our current focus has been on building up a set of basic collaborative capabilities and on improving our persistence tools. The PCCE environment now provides a secure persistent space that allows participants to locate each other, use asynchronous and synchronous messaging, collaboratively define and interact with computational workflows, share editing of documents, share progress and results, and hold videoconferences using the Access Grid. The next steps will be improving the integration of the tools and further exploring the security and usage model.

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

Enabling Group Communication within Distributed Science Applications

D. Agarwal*, K. Berket, O. Chevassut, and G. Egles, Lawrence Berkeley National Laboratory

Summary

Collaborative and distributed DOE applications need support for secure communication among members. This project has been developing and implementing the components necessary to provide reliable and secure group communication to support these applications. We have developed protocols using formal methods to ensure correctness. The protocol implementations are then further tested using our Distributed Test Framework. A secure group communication capability has a broad range of uses within cybersecurity, collaboration, and distributed computing applications.

A number of DOE scientific computing applications, such as shared remote visualization, shared virtual reality, and collaborative remote control of instruments are utilizing distributed resources to enhance collaboration, solve computational problems, and boost overall system availability. These applications require coordination and information sharing amongst the distributed components. This needs to be done reliably, securely, and efficiently.

There are two primary components to a reliable and secure group communication (RSGC) system: a reliable group communication protocol and a security protocol. We have been conducting the research and development required to build both of these components. The reliable group communication protocol we have developed is called InterGroup. InterGroup provides reliable delivery of application messages, ordered delivery of application messages, and membership services to the application. The security protocols we have developed are called the Secure Group Layer (SGL). SGL provides authorization and access control, session key establishment, and authenticity, integrity and confidentiality of the group communication. SGL uses the reliability properties provided by the underlying InterGroup system. A key aspect of the InterGroup and SGL protocols is that they were both designed using formal methods to scale to the Internet. This has allowed us to provide good performance along with very strong guarantees regarding the security and group communication properties they provide.

The most difficult tasks in developing group communication protocols are providing formal specifications and representations of the algorithms, and then proving that the algorithms meet the specifications. We use I/O automata to formally represent the specifications and algorithms of RSGC. The system properties are directly derived from the specifications and proven to hold in the algorithms by using established proof techniques. In FY2005 we completed work on a new set of algorithms and proofs that enhance the scalability of InterGroup. We

* Phone: 510-486-7078, E-mail: DAAgarwal@lbl.gov

then implemented these algorithms and included them in the RSGC software.

In the peer-group security model the participants are treated as equals. The challenge in the development of this security model resides in the cryptographic technologies to support it. Cryptographic algorithms for group Diffie-Hellman key exchange are necessary so that the group participants can exchange a session key as peers. In this project we have developed the group key exchange algorithms required to build a shared key through exchanges between the peers. We have also quantified the strength of these cryptographic algorithms via a rigorous treatment in the framework of provable security. This involves capturing the adversary's capabilities in adequate threat models and defining stringent security definitions. The proofs then show that our group Diffie-Hellman key-exchange algorithms achieve the security definitions under reasonable mathematical assumptions. In FY2005 we completed an initial implementation of SGL based on this theoretical foundation.

We used a modular approach in developing InterGroup. Each module implements a small part of the functionality of the system such as network detection, membership, or message ordering. The specifications and algorithms can then be developed on a per module basis. This approach also allows us to secure the GCS by securing the modules individually. In FY2005, we began work on integrating the SGL algorithms into this approach. We extend the InterGroup component algorithms with the SGL algorithms to meet the required security conditions. Then, we show that the addition of security does not break the system properties.

Testing of a distributed protocol is difficult, but essential. During FY2005, we developed a Distributed Testing Framework (DTF) to assist in the testing of our group communication protocols, The DTF provides an easy to use language to describe the initial conditions, executables to run, and external inputs to the system as a script. When DTF is running, there is a conductor process coordinating all of the activities and a small client process babysitting each individual instance of the distributed computation. DTF is designed to allow testing in any distributed environment. We are using DTF to test InterGroup and SGL on the Emulab testbed and on the Internet through PlanetLab. Using DTF, we are able to test virtually any aspects of the RSGC protocol in a distributed manner.

More information on this project, including publications and source code, can be found at http://www.dsd.lbl.gov/GroupComm/

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov

A Scalable and Secure Peer-to-Peer Information Sharing Tool

Karlo Berket*, LBNL
Abdelilah Essiari, LBNL
Artur Muratas, LBNL

Summary

We have developed a peer-to-peer system to support secure, location independent information sharing in the scientific community. In our system the owners of information are able to control how the information is stored, managed, and shared. This system allows seamless and secure sharing of information between multiple collaborators and allows informal collaborations to be formed quickly and securely.

Groups collaborating on scientific experiments have a need to securely share information and data. This information and data is often represented in the form of files. In a typical scientific collaboration, there are many different locations where data would naturally be stored. This makes it difficult for collaborators to find and access the information they need. Our goal was to create a lightweight file-sharing system that makes it easy for collaborators to find and use the data they need. Additionally, this system needed to be easy-to-use, easy-to-administer, and secure.

Collaboration tools and environments provide a set of persistent services to users. However, they often rely on a centralized infrastructure. This makes the tools unusable when a specific resource, such as a server is unavailable. Ideally, the collaboration environment should not depend on any specific resource; instead, the presence of resources should add value to the system. A collaboration environment should be structured to support informal, spontaneous collaborations as well as highly structured environments. Using on-line tools, it should be easy to begin collaborating, and incrementally add users and services as needed.

A key distinguishing aspect of our information-sharing tool is that we designed every component for such an environment. We use group communication, in particular the InterGroup protocols, to reliably deliver each query to all of the current participants in a scalable manner, without having to discover all of their identities. We use the Secure Group Layer (SGL) and Akenti to provide security to the participants of our environment. SGL provides confidentiality, integrity, authenticity, and authorization enforcement for the InterGroup protocols. Akenti provides access control to other resources.

During FY 2005 we have continued our research and development efforts. Our research efforts have focused on security in unstructured peer-to-peer systems. We have specified a security framework for dynamic collaborations that uses the public key infrastructure widely employed within the DOE scientific community [1].

* Phone: 510-486-4807, E-mail: Kberket@lbl.gov

Our research was inspired by the widely held belief that computer security is a hindrance to agile collaborations and social networks. A number of existing technologies provide secure and efficient solutions for well-established collaborations. However, when the collaboration is informal or dynamic these solutions fail by compromising security or efficiency. The research performed as part of this project [1][2] has shown that this does not have to be the case.

Our development efforts have focused on providing a secure, reliable, and efficient implementation of the information-sharing tool, called *scishare*. The implementation consists of a file-sharing specific application, a graphical user interface, and an authorization component. The file-sharing specific component of scishare provides a programmatic interface and command line tools for information sharing. This allows users to interact with the system without a graphical interface and developers to customize a user interface to their requirements.

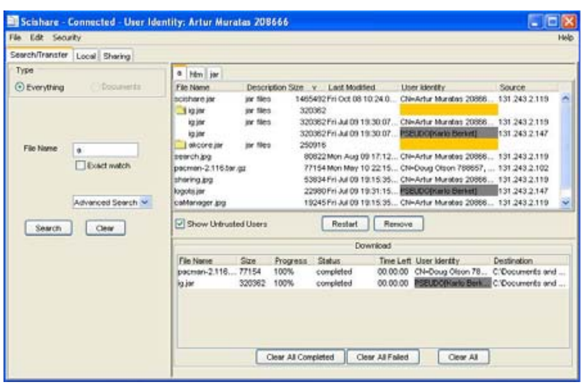


Figure 1. Performing a search in the scishare application.

The graphical user interface provides the user with access to the full functionality of scishare. The user can search for and download files (Figure 1), monitor the transfer of files, and manage locally-shared files (Figure 2).

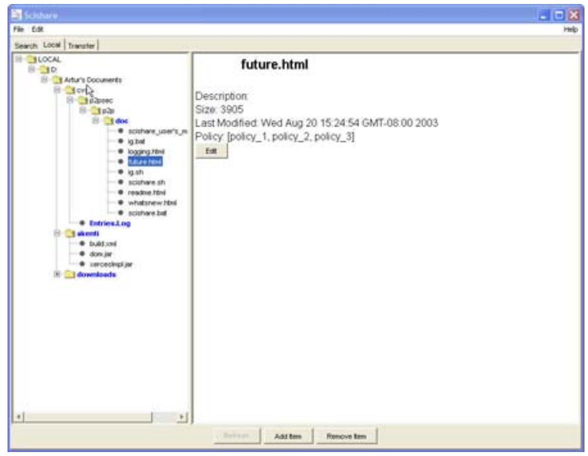


Figure 2. Managing locally-shared files in the scishare application.

The authorization component allows the user to specify fine-grained policies for access control to locally-shared files and provides enforcement of these policies.

The scishare software and source code are available for download at http://www.dsd.lbl.gov/scishare.

[1] Securing Resources in Collaborative Environments: A Peer-to-peer Approach. K. Berket, A.Essiari and M. R. Thompson, Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and Systems, Phoenix, AZ, Nov. 14-16, 2005. LBNL-58867.

[2] PKI-Based Security for Peer-to-Peer Information Sharing. K. Berket, A.Essiari and A. Muratas, Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing, Zurich, Switzerland, Aug. 25-27, 2004. LBNL-54975.

For further information on this subject contact:
Dr. Mary Anne Scott, Program Manager for Collaboratory Research
Mathematical, Information, and Computational Sciences Division
Office of Advanced Scientific Computing Research
Phone: 301-903-6368
scott@er.doe.gov